いずれかのクラスタサーバが起動したとき、マスタノード（サーバ）は最初にクラスタを形成する必要があります。Cluster Volume Manager を実行しているクラスタノードは、起動プロセスで Cluster Volume Manager のボリューム再構成デーモンを起動する必要があります。このデーモンは、クラスタ共有ディスクグループを検出すると、それらのディスクグループを自動的にインポートし、それらのボリュームをノードからアクセスできるようにします。クラスタ共有ディスクグループを最初にインポートしたノードは、そのグループのマスタノードになります。追加のサーバがクラスタに参加した場合、それらのサーバはクラスタ共有ディスクグループをインポートし、そのディスクグループのクライアントユーザーになります。Cluster Volume Manager を備えたクラスタ内のすべてのサーバは、起動時にすべてのクラスタ共有ディスクグループをインポートします。
クラスタ共有ディスクグループのマスタノードは、ディスクグループのすべてのメタデータ変更を行います。メタデータ変更には、たとえば、ディスクグループ内部のストレージ容量を使用したボリュームの作成、ボリュームのサイズ変更、ミラー化ボリュームからのミラーコピーの削除などが含まれます。ディスクグループのメタデータを変更した場合、マスタは必ず分散トランザクションの手法を使用してすべてのノードの同期をとり、変更がクラスタ全体で同時に有効となるようにします。

Before one of the various cluster servers start up, the master node (server) needs to form a cluster first. The cluster node that is running the Cluster Volume Manager needs to start up the volume reconfiguration daemon as an active process in the Volume containing the Cluster Volume Manager. When this daemon detects the cluster sharing disc group, it imports them and enable the access of these volumes from the node automatically. The node that first imports the cluster sharing disc group becomes the master node for the group. If an additional server joins the cluster, the cluster server imports the cluster sharing disc group which becomes a client user of the existing disc group. All the servers that have Cluster Volume Manager in the cluster imports all cluster sharing disc groups when they start up.
The master node for the cluster sharing disc group executes all metadata changes in the disc group. The following are examples of metadata changes; volume creation that uses the disc group's storage capacity, volume size change, mirror copy deletion from a mirrored volume, etc. In case where the metadata changes in the disc group, the disk group master always initiates the synchronization of all the nodes using a decentralized transaction method and makes the change available to the whole cluster at the same time.

WHITE PAPER

VERITAS

VERITAS Cluster Volume Manager
VERITAS Cluster File System

クラスタ環境のための新しい
VERITAS のボリューム管理
およびファイルシステム
テクノロジ

VERITAS

目次

図

Ⅰ. VERITAS Cluster Volume Manager
共有ボリュームテクノロジー

VERITAS Volume Manager

VERITAS Volume Manager は、ディスクまたはハードウェアベースの RAID アレイを、障害に強くパフォーマンスの高い柔軟な論理ボリュームに集約することにより、オンラインストレージのアベイラビリティとパフォーマンスを向上させます。

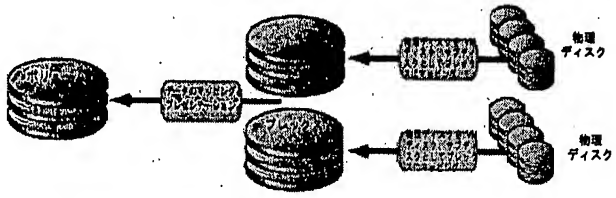


図 1 : VERITAS Volume Manager のボリュームアーキテクチャ

Volume Manager は、図 1 に示すような階層的なストレージオブジェクトアーキテクチャによるボリューム管理を提供します。物理ディスク上の各ブロック範囲は、プレックスに集約されます。それぞれのプレックスは、耐障害性の高い（たとえば RAID）とデータマッピング（たとえばストライプ化）の特性を備えています。プレックスは、さらにボリュームに集約され、ボリュームはファイルシステム、データベース、およびアプリケーション等によりディスクとまったく同じように使用されます。

表 1 : Volume Manager のボリュームタイプ

ボリュームタイプ	ディスク耐障害性の高い（データのアベイラビリティ）	パフォーマンス（単一ディスクとの比較）	エンタープライズ IT でのアプリケーション
n ウェイのミラー化（RAID 1 とも呼ばれる）	非常に高い	高い読取りパフォーマンス、同等の書込みパフォーマンス	小さな（単一ディスクの）クリティカルファイル
ストライプ化および n ウェイのミラー化	非常に高い	非常に高い読取りパフォーマンス、高い書込みパフォーマンス	大きな（マルチディスクの）クリティカルファイル
クリティカルなストライプ化（RAID 5）	高い	高い読取りパフォーマンス、低い書込みパフォーマンス	重要な「ほとんど読取り専用」のデータ
ストライプなどのストライプ化（RAID 0 とも呼ばれる）	単一ディスクより低い	高い読取りおよび書込みパフォーマンス	置き換えが簡単だがパフォーマンス上クリティカルなデータ
連結（容量集約型）	単一ディスクより低い	同等の読取りおよび書込みパフォーマンス	置き換えが簡単だがパフォーマンス上クリティカルなデータ

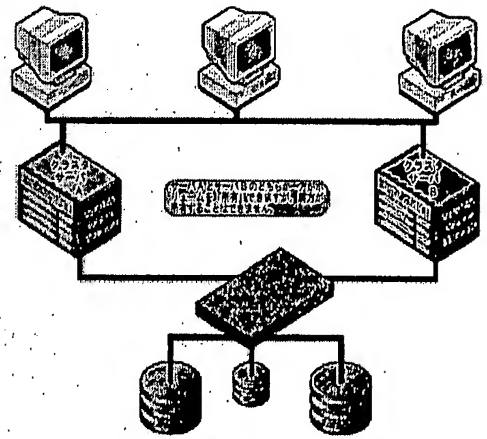
表 1 は、Volume Manager がサポートするボリュームタイプを示します。クラスタコンピューティングの大きな特徴である耐障害性の向上をはかるために、n ウェイのミラー化ボリュームと RAID ボリュームの両方がサポートされています。

ミラー化ボリュームは、ディスクの障害に対して最も安全な構成となります。また、Volume Manager は、三つ以上の同一のデータコピーを別々のディスクに持つミラー化ボリュームをサポートしています。3 ウェイのミラー化は、クリティカルな業務データの凍結イメージがバックアップや新規アプリケーションのテストに必要な場合に便利です。3 ウェイのミラー化ボリュームから一つのコピーをバックアップまたはテストのために切り離しても、稼働中のアプリケーションは、耐障害性の高い 2 ウェイのミラーによりデータで処理を続行できます。

Volume Manager は複数のディスクまたはディスクプレックスにわたりデータのストライプ化を行うこともでき、それにより I/O の負荷が均等化され、パフォーマンスが最適化されます。ストライプ化を単独で、あるいはミラー化やパリティ RAID との併用により、耐障害性の高いハイパフォーマンスボリュームを実現できます。

また、Volume Manager はストレージの容量管理に新たな柔軟性を提供します。Volume Manager のボリュームは、システムをオンラインの状態にしたまま拡張することができ、システム管理者は予定外のストレージ要求に対しても、アプリケーションをほとんど、あるいはまったく中断せずに対応できます。

VERITAS Volume Manager は、数千に及ぶ企業 IT システムに導入され、データのアベイラビリティ、パフォーマンス、および管理の柔軟性を高めてきました。Volume Manager の基本設計は、ボリュームごとに単一の制御ポイントを設定する、すなわち、一つのボリュームは一つのホストコンピュータのみにより管理され、使用されるようにするということです。このモデルは、図 2 に示すように、VERITAS FirstWatch™ および VERITAS Cluster Server の非共有データクラスタで使用され、多くのユーザーに利用され支持されてきました。



VERITAS Volume Manager の耐障害性の高いボリューム

図 2 : VERITAS Cluster Server と Volume Manager ボリュームによる構成

図 2 の耐障害性の高いボリュームは、ある時点では、一つのサーバにのみアクセスは限定されています。このボリュームを使用しているアプリケーションまたはサーバに障害が起きた場合、クラスタサーバはアプリケーションのフェイルオーバー先となるサーバへボリュームの所有権を移行します。

VERITAS Cluster Volume Manager

VERITAS Volume Manager は実用上きわめて便利であることが実証されていますが、その単一制御ポイントモデルは、共有データクラスタをサポートしていません。このため、ベリタスソフトウェア では VERITAS Volume Manager をベースにした Cluster Volume Manager を開発し、VERITAS Cluster Server またはその他（Oracle Parallel Server など）の共有データクラスタ用に堅固な 共有ボリュームテクノロジーを提供しています。

Cluster Volume Manager は、Volume Manager をベースとした製品で、次の機能が追加されています。（SANPoint Foundation Suite など、いくつかの製品に組み込まれて提供されます。）

- 複数のサーバから各ボリュームへの同時アクセス
- クラスタ全体の論理デバイスのネーミング
- すべてのサーバで整合性のある、ボリューム状態の論理ビュー
- クラスタ内の任意のサーバからのボリューム管理
- サーバに障害が起きた後、アクセス可能で残ったボリュームへの生き残ったサーバからのアクセス
- ボリュームフェイルオーバーのないアプリケーションフェイルオーバー

Volume Manager と同様に、Cluster Volume Manager は、物理ディスクと、ハードウェア RAID アレイサブシステムによりエクスポートされた仮想ディスクの両方を管理できます。

Cluster Volume Manager のアーキテクチャの概念

Volume Manager と同様に、Cluster Volume Manager はディスクをディスクグループに編成します。それぞれのボリュームは、一つのディスクグループに属するディスクから割り当てられます。Cluster Volume Manager のディスクグループであるボリュームは、プライベート、あるいはクラスタ共有の属性を持ちます。ボリュームが物理的にクラスタ全体に接続されていても、プライベートディスクグループに属するディスクは 1 台のサーバだけからアクセスできます。図 3 の場合、A、B、C、D の各ディスクグループは、それぞれが単一のサーバに接続されているので、必然的にプライベートディスクグループとなります。各サーバは、サーバごとに一つのプライベートのルートディスクグループを持つ必要があり、そのことは、そのグループに属するディスクが物理的に複数のサーバに接続されているかどうかとは関係ありません。

BEST AVAILABLE CO

VERITAS

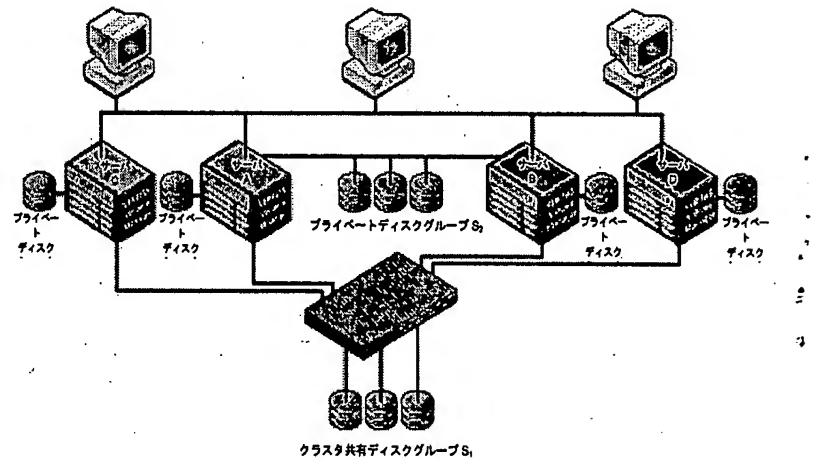


図 3 : プライベートディスクグループとクラスタ共有可能ディスクグループ

クラスタ共有ディスクグループは、複数のサーバから同時にアクセスできます。これらのディスクグループは、物理的にクラスタ内のすべてのサーバに接続されている必要があります。図 3 の場合、ディスクグループ S1 はクラスタ共有ディスクグループにすることができます。ディスクグループ S2 はサーバ A とサーバ B だけがアクセスできるので、クラスタ共有にできません。ディスクグループ S2 は、サーバ A とサーバ B のどちらかが所有するプライベートディスクグループとして指定できます。

Cluster Volume Manager は、単純ボリューム（単一ディスク）、スパンボリューム（連結）、ミラー化ボリューム、ストライプ化したミラー化ボリューム、およびミラー化したストライプ化ボリュームをサポートします。今後、パリティ RAID ボリュームのサポートを予定しています。（時期未定）

クラスタ共有ボリュームサーバとクライアントノード

VERITAS Volume Manager の場合と同様に、システム管理者は vxdg ユーティリティを使用してディスクグループをプライベートとクラスタ共有のどちらかに指定します。プライベートディスクグループは、機能的には単一ホストシステム上の VERITAS Volume Manager ディスクグループと同じものです。

いずれかのクラスタサーバが起動したとき、マスタノード（サーバ）は最初にクラスタを形成する必要があります。Cluster Volume Manager を実行しているクラスタノードは、起動プロセスで Cluster Volume Manager のボリューム再構成デーモンを起動する必要があります。このデーモンは、クラスタ共有ディスクグループを検出すると、それらのディスクグループを自動的にインポートし、それらのボリュームをノードからアクセスできるようにします。クラスタ共有ディスクグループを最初にインポートしたノードは、そのグループのマスタノードになります。追加のサーバがクラスタに参加した場合、それらのサーバはクラスタ共有ディスクグループをインポートし、そのディスクグループのクライアントユーザーになります。Cluster Volume Manager を備えたクラスタ内のすべてのサーバは、起動時にすべてのクラスタ共有ディスク

Copyright 2002 VERITAS Software Corporation. All rights reserved. VERITAS, VERITAS SOFTWARE, VERITAS の、VERITAS SANPoint Foundation Suite は、米国および
国の VERITAS Software Corporation の登録商標です。その他の名称、製品名等、それぞれの会社の商標または登録商標です。
本資料の仕様、記載内容等は予告なく変更する場合がありますので、ご了承ください。

グループをインポートします。
クラスタ共有ディスクグループのマスタノードは、ディスクグループのすべてのメタデータ変更を行います。メタデータ変更には、たとえば、ディスクグループ内部のストレージ容量を使用したボリュームの作成、ボリュームのサイズ変更、ミラー化ボリュームからのミラーコピーの削除などが含まれます。ディスクグループのメタデータを変更した場合、マスタは必ず分散トランザクションの手法を使用してすべてのノードの同期をとり、変更がクラスタ全体で同時に有効となるようにします。

Cluster Volume Manager の機能の概要

Cluster Volume Manager のリリース 3.2 は、最大 16 台のサーバ間でのボリュームの共有（つまり 16 ノードのクラスタ）をサポートしています。Cluster Volume Manager は、複数のパス（I/O パス）上にあるディスクアレイにアクセスできる RAID コントローラや、すべてのクラスタノードのアクセスパスを同時に切り替えることができる RAID コントローラを認識できます。「ローリング」ソフトウェアアップグレードがサポートされるので、Cluster Volume Manager のインストレーションは、一度に一つのクラスタノードをサービスから外すだけで更新できます。

クラスタボリュームの意味

Cluster Volume Manager は、名前が示すとおり、クラスタボリュームマネージャです。Cluster Volume Manager はボリュームを、ディスクと同じように、クラスタファイルシステムやその他のシステムアプリケーション、たとえば分散データベースマネージャなどで使用するためにエクスポートします。Cluster Volume Manager はメタデータの更新の同期をとり、クラスタ共有ディスクグループのボリュームについて、クラスタ全体で整合性のあるビューを提供します。しかし、ボリューム自体は、アプリケーションからの読取りコマンドや書込みコマンドに対してディスクのように動作します。

ボリュームは、ディスクのように、ホストの読取りコマンドと書込みコマンドを無差別に実行します。サーバは、ディスクに物理的にアクセスできる場合、ディスクに対して、ボリュームの場合と同じように、有効なすべての読取りコマンドまたは書込みコマンドを発行できます。この動作は、一つのディスクまたはボリュームが、そのボリュームへのすべてのアクセスを調整する一つのファイルシステムまたはデータベースインスタンスによりマウントされる単一サーバシステムでは、適切な動作です。しかし、クラスタでは、複数のノードが同時に一つのディスクの読書きを行う場合があります。ノード間の調整を行わないと、データの更新が誤って相互に上書きされるおそれがあり、同じブロック範囲へ別のノードが書込みを行っているときに読取りを実行すると、整合しない「移動中の」データイメージが返ってくるおそれがあります。

Cluster Volume Manager は、上記のように、このディスクとして読込みと書込みを実行します。しかし、Cluster Volume Manager は、異なるクラスタノードのユーザーによるデータの読書きが相互に干渉しないことを保証するわけではありません。そのような保証は、ボリュームを使用するファイルシステムまたはデータベースマネージャが行う必要があります。Cluster Volume Manager は、クラスタに対応したデータマネージャ、たとえば、VERITAS Cluster File System などと併用するように設計されています。このデータマネージャは、ファイルやディスクブロックなど、クラスタ全体での共有データオブジェクトへのアクセスを調整する分散ロックマネージャを備えています。Cluster File System は、VERITAS Global Lock Manager を使用してデータアクセスを調整します。

Cluster Volume Manager とシステムの障害

Cluster Volume Manager は、VERITAS Volume Manager テクノロジを使用して二つ以上のディスクにデータのミラー化を行い、ディスクの耐障害性の向上させます。ボリュームをミラー化すると、Volume Manager はアプリケーションの書込みを、そのボリュームに対応するそれぞれのディスクのブロックに対して行います。

サーバは、ミラー化が進行中のボリュームへの書込みに失敗することもあります。それにより、更新されたコピーとそうでないコピーが生じ、ボリュームの内容が整合しなくなることも考えられます。

制御ポイント（ボリュームマネージャを実行しているサーバ）がクラッシュした後に耐障害性の高いボリュームの内容の整合性を保証することは、ボリューム管理テクノロジでは既知の問題であり、解決方法もいくつか知られています。たとえば、ミラー化ボリュームをクラッシュ後に整合させるには、一つのディスクをマスタとして指定し、そのすべての内容を、一緒にミラー化された各ディスクにコピーします。しかし、このような解決方法は時間がかかり、データとアプリケーション双方のアベイラビリティを低下させます。

ベリタスソフトウェアの製品は、システムがクラッシュした後、ボリュームとファイルシステムの整合性を迅速に復元できることでよく知られています。ミラー化ボリュームの場合、DRL（ダーティリージョンログ）は、ボリュームのどのブロック範囲（「リージョン」）が最近更新されたかを示し、その結果、システムクラッシュの後、どのブロック範囲が整合していない可能があるかを示します。DRL のエントリは、ミラー化ボリュームのディスクが更新される前に書込まれます。あるボリュームのすべてのミラーが更新された場合、そのリージョンは、もはや危険でないものとしてマークされます。オーバヘッドを最小限に抑えるため、このログへの書込みは、保留中の別の更新がそれを必要とするまで行われません。

Cluster Volume Manager は、ノードのクラッシュからリカバリすると DRL を読取り、危険のフラグが付いているミラー化ボリュームリージョン内のデータを再コピーします。ある時点で危険となるボリュームのリージョンは小さな断片にすぎないと考えられるので、リカバリ時間は、非常に大きなボリュームでも、強制的なコピーの場合（数分または数時間）よりずっと短く、数秒で済みます。

Ⅱ. VERITAS Cluster File System

クラスタのファイルシステム

VERITAS Cluster Volume Manager は、クラスタ全体にわたり、堅固な論理ボリュームをデータマネージャと Raw デバイスアプリケーションからアクセスできるようにします。VERITAS Cluster File System を使用すると、一つのファイルシステムを複数のクラスタサーバで同時にマウントして使用でき、そのファイルシステムを使用するすべてのアプリケーションが同じサーバ上で実行されているのと同じことになります。図 4 は、クラスタ内の Cluster File System を示しています。

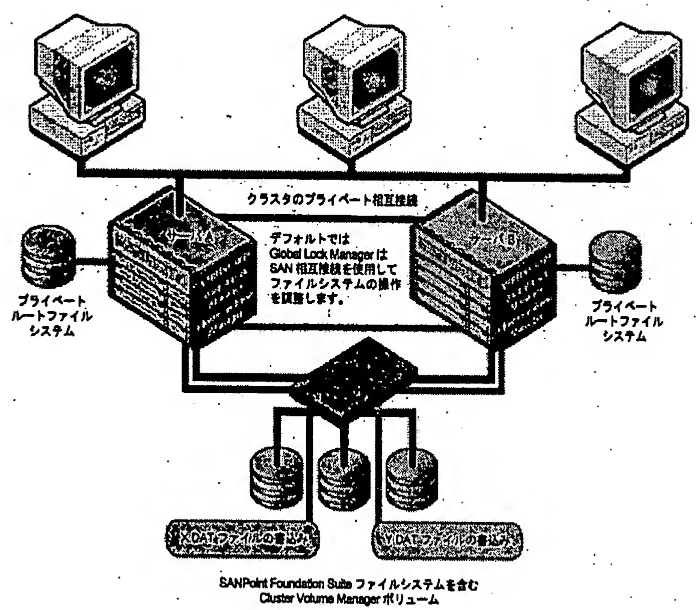


図 4 : Cluster Volume Manager 上に階層化された VERITAS Cluster File System

VERITAS

Cluster File System のプロパティ

VERITAS Cluster File System は、単一ホストの VERITAS File System をベースとしています。VERITAS File System は、その成熟度と豊富な機能セットにより、エンタープライズ UNIX 環境に特に推奨するファイルシステムです。VERITAS File System がエンタープライズ向けアプリケーションとして特に有効である理由は次のとおりです。

- テラバイトのサイズまでのファイルを簡潔にマップできるエクステントベースのスペース管理。
- ほとんどのシステムクラッシュからの迅速なリカバリ。これには、最近のファイルシステムメタデータの更新を追跡する自己クリア方式のインテントログが使用されます。
- ファイルシステムを使用しながら拡張とディフラグメントができるオンライン管理機能。
- Quick I/O 機能。これにより、対応するデータベースマネージャでファイルを Raw パーティションとして扱うことによりカーネルロックを迂回できます。また、QuickI/O の拡張機能である Cashed Quick I/O では 32 ビットアプリケーションで 4 ギガバイトを超えるシステムキャッシュを利用できます。（この機能は、Database Edition にて提供されます。）

VERITAS Cluster File System を使用すると、これらすべての機能をサーバのクラスタから使用できます。また、Cluster File System は、クラスタ環境を利用して次の機能も提供します。

- クラスタ全体にわたるファイルシステムの状態のスナップショットの作成。これにより、ファイルシステムの一貫したオンディスクイメージを必要とするオペレーション（たとえば、バックアップ用またはテスト用のミラー化ボリュームからコピーを削除するなど）をクラスタ環境で行うことができます。
- クラスタ全体とローカルの両方におけるファイルシステムのマウント。これにより、管理者はクラスタノード間でデータを共有したり、アプリケーション要件による制約を離れてデータを共有したりできます。
- Cluster File System 自体の「ローリング」更新。これにより、Cluster File System をノードごとに更新でき、アップグレードプロセス全体を通して、クラスタを一体のものとして操作できます。

Cluster File System ファイルシステムを使用しているサーバに障害が起きた場合、そのサーバのアプリケーションが、残存するサーバへフェイルオーバーする可能性がありますが、多くの場合、ファイルシステムの再起動は必要ありません。なぜなら、ファイルシステムは依然として稼働しているからです。このため、システムクラッシュの後にアプリケーションを再起動したとき、通常は時間消費の主要な原因となるアプリケーションデータのリカバリ作業を排除できます。

VERITAS Cluster File System の利点

SANPoint Foundation Suite と SANPoint Foundation Suite HA では、次に述べるように、ハードウェアの制限から生じる大規模システムの管理作業の多くが簡素化されるか不要になります。

- Cluster Volume Manager はテラバイトのボリュームの作成と管理ができるので、ファイルシステムをディスク容量の限度内に収まるようにパーティション分割する必要はほとんどありません。
- Cluster File System は 1 テラバイトまでの容量のファイルシステムをサポートできるので、パーティション分割が必要となるのは、非常に大量のデータを使用し、ファイルシステムのアドレス指定の制限のために分割が必要となる場合だけです。
- クラスタ内のすべてのサーバが Cluster File System のクラスタ共有可能ファイルシステムにアクセスできるので、参照データまたはアプリケーションのイメージとライブラリの整合性を複数のサーバにわたり維持することは自動で行われます。すべてのクラスタノードが、同じ参照データおよびイメージから作業を行うことができます。それだけでなく、非共有データクラスタでアプリケーションと参照データの複数の同一コピーに必要となるストレージの容量は、すべてのサーバが同じデータとイメージから作業を行うときは不要になります。
- すべてのサーバがすべてのファイルにアクセスできるので、アプリケーションを各サーバに割り当て、負荷を均等化したり、その他のオペレーション要件を満たすことができます。同様に、フェイルオーバーも、データのアクセス可能性により制約を受けないため、柔軟に行うことができます。
- すべてのクラスタノードが 各 Cluster File System のマスタになり得るので、ファイルシステムのマスタ権をクラスタノード間で分散することにより、フェイルオーバー時間に占めるファイルシステムのリカバリの部分を、n 個のノードからなるクラスタでは n 分の 1 に減少させることができます。
- エンタープライズ RAID サブシステムを投資に見合うものにし、より効果的に使用できます。なぜなら、それらのサブシステムのストレージ容量をすべてのサーバがマウントでき、変化するビジネスの要求に合わせて、ハードウェアの再構成ではなく管理オペレーションにより再割り当てができるからです。
- ファイルシステムを共有できることにより、ストライプ化の幅が広がり、より大きなボリュームが使用可能となり、アプリケーションの I/O ロードバランシングが向上します。それぞれのサーバの I/O 負荷がより多くの I/O リソースに分散されるだけでなく、Cluster File System 共有ファイルシステムを使用すると、すべてのサーバの負荷がすべての I/O リソース間で均等化されます。
- ボリュームとファイルシステムの数が少なくなるため、管理者、ユーザー、アプリケーションの開発者と管理者は個々のオブジェクトを見つけるのが容易になります。またバックアップについても同様です。
- サーバの追加によるクラスタの拡張が容易になります。なぜなら、それぞれの新規サーバのデータストレージ構成は、ほとんどの場合、事前定義されているからです。新規サーバは単に既存のクラスタ全体のボリュームとファイルシステムの構成を適用することにより可能になります。
- Cluster File System の単一システムイメージ管理モデルは、すべてのファイルシステム管理オペレーション（サイズ変更は除く）をそれらのオペレーションが呼び出された場所に依存しないものにする（ただし、実際のメタデータオペレーションは現行の Cluster File System マスタノードにより実行される）ので、管理を簡素化します。
- 管理者は、使用可能なサーバ容量と処理するデータがありながら、その二つを結び付ける方法がない

いという状況を回避できます。Cluster File System クラスタ共有可能ファイルシステムを使用すると、すべてのサーバが、実質的に単一サーバファイルシステムと同じように、すべてのデータにアクセスできます。

- SANPoint Foundation Suite および SANPoint Foundation Suite HA の新しいオプション機能である VERITAS FlashSnap™ により、データの分析、バックアップ、テストおよびレポート機能などのオペレーションを運用環境に影響を及ぼすことなく実行できます。

Cluster File System とアプリケーション

多くのアプリケーションが Cluster File System の恩恵を受けることができます。従来の「クラスタ非対応」アプリケーションも、クラスタ内のすべての場所で実行でき、すべての場所のデータにアクセスできます。マルチアプリケーションクラスタでは、Cluster File System ファイルシステムが比較的大きなボリューム上に構築されていれば、ロードバランシングの向上により全体的な I/O パフォーマンスが向上します。これらの利点は、Cluster File System をインストールすれば自動的に得られます。チューニングやその他の管理操作は必要ありません。

多くのアプリケーションはパーティション分割が可能です。つまり、それらは複数の同時実行スレッドから構成され、それらのスレッドは、それぞれのデータアクセスを調整する方法があれば、異なるサーバ上でも実行できます。Cluster File System は、その調整を行います。そのようなアプリケーションは「クラスタ対応」にすることができ、インスタンスを連携させてクライアントとデータアクセスの負荷を均等化し、それにより、単一サーバの容量を超えた拡張を行うことができます。そのようなアプリケーションでは、Cluster File System は I/O の負荷を均等化するだけでなく、共有データアクセスを提供することにより、クラスタノード間でアプリケーションレベルのロードバランシングを可能にします。

VERITAS Cluster File System のアーキテクチャ

サーバ/クライアントのファイルシステム設計

VERITAS Cluster File System は、マスタ/クライアントのアーキテクチャを使用し、共有ボリューム上のファイルシステムメタデータを管理します。それぞれの Cluster File System ファイルシステムを最初にマウントしたサーバが、そのファイルシステムのマスタになります。クラスタ内のその他のノードは、すべてクライアントになります。アプリケーションは、図 5 に示すように、そのアプリケーションを実行しているサーバから直接、ファイル内のユーザーデータにアクセスします。しかし、Cluster File System ファイルシステムメタデータを更新できるのは、ファイルシステムのマスタノード（そのファイルシステムを最初にマウントしたノード）だけです。Cluster File System マスタノードはすべてのメタデータの更新に責任を負い、ファイルシステムのメタデータ更新インテントログの保守にも責任を負います。クライアントサーバは、ファイルシステムメタデータを更新する必要がある場合（たとえば、新しいファイルを割り当てたり古いファイルを削除する必要がある場合）、マスタにその要求を伝え、マスタは実際の更新を行い、要求元のサーバに応答します。この設計により、ファイルシステムメタデータの整合性と、システムクラッシュからのリカバリに使用されるインテントログの整合性が保証されます。

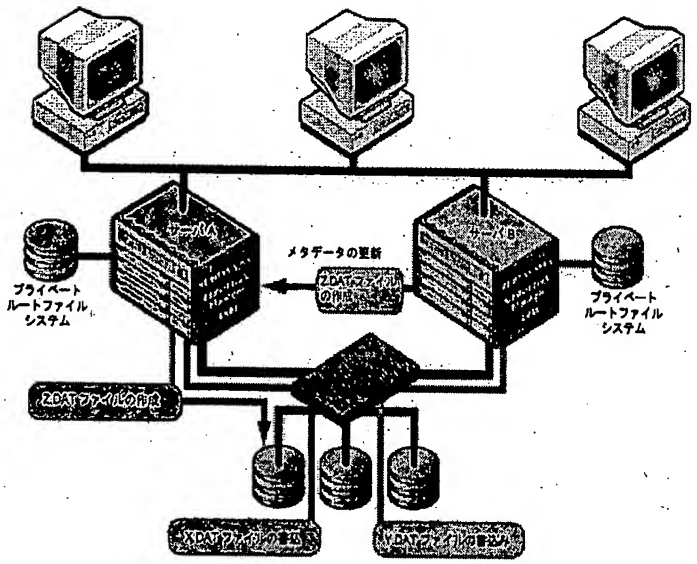


図 5 : VERITAS Cluster File System のサーバクライアントアーキテクチャ

Cluster File System の耐障害性について

Cluster File System マスタサーバに障害が起きた場合、残存するクラスタノードは新しいマスタを選出します。新しいマスタはファイルシステムのインテントログを読取り、障害が起きたときに処理中だったメタデータの更新があれば、それを完了します。アプリケーション I/O は、リカバリの間、停止され、通常ではリカバリまでに数秒かかります。ファイルシステムは、再び整合した状態になると新しいマスタとクライアントにより自動的にマウントされ、アプリケーションの処理が再開されます。

Cluster File System クライアントノードはファイルシステムメタデータを直接更新しないので、Cluster File System クライアントノードに障害が起きても、メタデータの修復は必要ありません。したがって、クライアントノードの障害からの Cluster File System のリカバリは、マスタノードの障害からのリカバリに比べ、高速になります。

Cluster File System と VERITAS Global Lock Manager

Cluster File System は、クラスタで UNIX 単一ホストファイルシステムとして動作するために VERITAS Global Lock Manager を使用します。最も重要なのは、書込み動作です。UNIX ファイルシステムでは、あるアプリケーションでファイルにデータを書込んだ場合、後続のアプリケーションで同じファイル領域に対して読取りを行うと、新しいデータが取り出されます。これは、そのデータがファイルシステムによりキャッシュされているだけでディスクに書込まれていない場合でも同じです。アプリケーションで「古い」データが取り出されたり、前の書込み結果の一部分だけが取り出されたりすることはありません。

単一ホストシステムでは、このファイルシステムが制御の中心となり、キャッシュ内にどのようなデータが入っているかを考慮して動作しています。クラスタでは、Cluster File System クラスタ共有可能ファイルシステムをマウントしたノードは、それぞれ独自のキャッシュを備えています。Cluster File System においても単一ホストの書込みと同様に動作するために、各キャッシュ間の整合性を保ち、それぞれのキャッシュがアプリケーションによるキャッシュデータの更新を、そのデータの出所がどのクラスタノードであるかに関係なく、即座に反映する必要があります。

Cluster File System は、ファイル内の一つのブロック範囲を読書きする前に、Global Lock Manager を使用してファイルをロックし、クラスタ内の別のノードがそのブロック範囲を同時に更新したり、更新の完了前に読取ったりしないようにします。

Global Lock Manager は、オブジェクトをロックする場合、最初にロックのマスタノードを選出します。Global Lock Manager はロック要求をそのマスタ（要求を行ったノードの場合もある）へ送り、マスタはそれに対する応答として、ロックを認めるか、クラスタ内の他の場所ですでにロックが存在する場合はそのロックを無効にしようとします。

ロックが最終的に認められた場合、Cluster File System は要求された I/O を実行し、最後に、データがディスクへ書込まれた時点でロックを解放します。アプリケーションから見た場合、Global Lock Manager はローカルファイルシステムのロックメカニズムのように、見えないところで機能します。アプリケーションは Global Lock Manager を認識できなくてもかまいません。

同時に同じデータに対してオペレーションを行う各アプリケーションは、それぞれが管理する、より高いレベルのデータオブジェクト用に、独自のロックメカニズムを備えている必要があります。

Cluster File System は、GAB（Group Membership and Atomic Broadcast）プロトコル（図 6 を参照）を通してクラスタの状態の推移（クラスタから出たり、クラスタに参加したりするノード）を認識します。サーバに障害が起きた場合、または新しいサーバがクラスタに参加した場合、Cluster File System は既存のすべてのロックについて再マスタ指定（新しいマスタの選出）を行います。クラスタの推移前に存在していたすべてのロックが再マスタ指定された時点で、アプリケーションの処理が再開されます。アプリケーションから見た場合、I/O 要求はクラスタの推移の間とロックが再マスタ指定されている間、停止されますが、それ以外、この推移は見えないところで行われます。

Cluster File System と VERITAS Cluster Server のプロトコル

図 6 は、VERITAS Cluster File System とインタフェースする主要な機能コンポーネントを示しています。さまざまなクラスタノードからのファイルへのアクセスを制御する VERITAS Global Lock Manager の役割については、すでに説明した通りです。

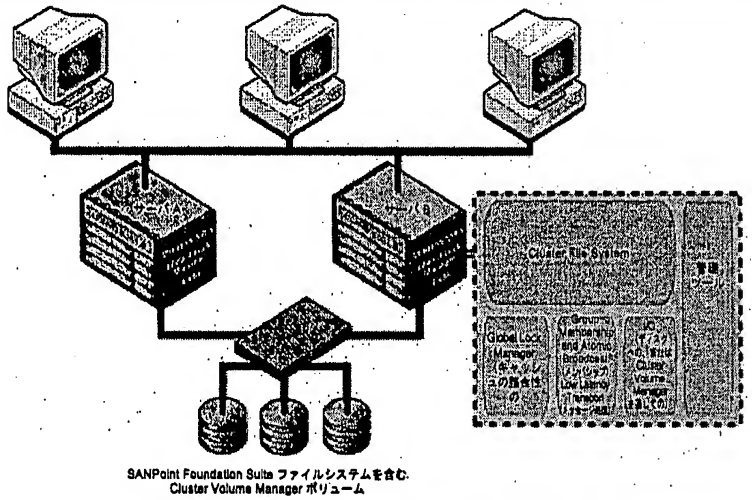


図 6 : VERITAS Cluster File System に統合されているコンポーネント

Cluster File System は、Cluster File System がメンバシップの判別とノード間通信のために、GAB（Group Membership and Atomic Broadcast）プロトコルと LLT（Low Latency Transport）プロトコルという 2 つのプロトコルを使用します。GAB と LLT は VERITAS Cluster Server 固有のプロトコルで、ファイバチャネルトランスポート（オーバヘッドと待ち時間を最小限に抑えるため）またはイーサネットデータリンクプロトコル上に直接導入できます。

GAB は、クラスタ全体にブロードキャストされたメッセージが正しく受信され（つまり確認され）、すべてのノードにより同じ順序で受信されることを保証するという意味で、一つのブロードキャストプロトコルです。GAB の主な用途はメンバシップサービスを提供することで、クラスタ全体に提供するだけでなく、Cluster File System インスタンスなど、メンバシップサービスを実行するアプリケーショングループにも提供します。GAB のメンバシップサービスを使用すると、起動とシャットダウンを正しい順序で行うことができます。

LLT は、その名が示すように、多数の小さなメッセージ（分散ロックトラフィックの特徴）を単純なネットワークトポロジ（クラスタの一般的な特徴）で高速配信するために最適化されています。

GAB と LLT は、どちらもクラスタ内のすべてのサーバを接続した冗長データリンクで動作するように設計されています。Cluster Server は、一つの通信リンクの障害によりクラスタの分割が起きる可能性を最小限に抑えるために、冗長クラスタ通信リンクを必要とします。

VERITAS

ベリタスソフトウェア株式会社
〒100-0011 東京都千代田区内幸町2丁目2番2号 富国生命ビル
www.veritas.com/jp

Copyright 2002 VERITAS Software Corporation. All rights reserved. VERITAS、VERITAS SOFTWARE、VERITAS は、VERITAS Software Corporation の米国およびその他の国における登録商標です。その他の会社名、製品名は、それぞれ各社の商標または登録商標です。
仕様は予告なく変更することがありますので、ご了承ください。